REMARKS

Claims 1-22 are pending. Claims 1, 12 and 21 are independent.

Rejections under 35 U.S.C. § 101

Applicant acknowledges with thanks the examiner's withdrawal of the rejections of claim 21 under 35 U.S.C. § 101.

The examiner maintained his rejection of claim under 35 U.S.C. §101 on the ground that the claimed invention is directed to non-statutory subject matter.

Specifically, the examiner states:

> 11. As to a) above, applicant clearly taught that the his multithreaded processor (12,20) can use freeware available over the internet (see page 2, lines 10-15). While the multithreaded processor can be a hardware, the freeware available over the internet is not. It is not sure what applicant intended to cover. The method of maintaining execution of multithreaded in multithreaded processor could be a mere program without a computer readable storage storing the threads. Furthermore, the register set without a functional descriptive material stored therein is not statutory since no requisite functionality is present to satisfy the practical application requirement (see discussion already set forth in page 2 of the last Office action). (Page 4 of the Office Action)

Applicant amended independent claim 1 to replace the wording "an executing thread" with "a thread executing" to clarify that independent claim 1 is directed to "accessing, ..., a register set ..." , and not to executing a thread. Applicant's independent claim 1 thus recites that the accessing operation is performed in a multithreaded processor including a register set. A multithreaded processor and a register set are both tangible elements. Thus, claim 1 recites a concrete, useful and tangible action, which is all that is necessary to make a claim statutory. Therefore, applicant's independent claim 1 is statutory.

Provisional Double Patenting Rejections

The examiner continues to provisionally reject claims 1, 12 and 21 under the judicially created doctrine of obviousness double patenting as being unpatentable over claims 18 and 2, respectively, of copending application No. 09/760,509.

Applicant will hold the examiner's provisional double patenting rejections of claims 1, 12 and 21 in abeyance upon an indication of allowable subject matter.

Rejections under 35 U.S.C. §102

The examiner maintained his rejection of claim 1 under 35 U.S.C. §102(b) as being anticipated by Cheng et al. ("The Compiler for Supporting Mutlithreading in Cyclic Register Windows", 1996). The examiner also maintained his rejections of claims 1-21 under 35 U.S.C. §102(a) as being anticipated by U.S. Patent No. 5,870,597 to Panwar et al.

Applicant amended independent claim 1 to include features similar to the features previously recited in claims 9 and 10, namely, that accessing absolutely any one of the relatively and absolutely addressable registers comprises providing the exact address of the register, the exact address specified in an instruction associated with the thread. Applicant similarly amended independent claims 12 and 21. Further, applicant amended claim 10 to make it dependent from independent claim 1 and to make the claim language consistent with the language of claim 1. Applicant cancelled claims 9 and 22.

In response to applicant's Amendment in Reply to Action of August 29, 2005, the examiner stated:

> 12. As to b), e) above, applicant is reminded that unclaimed features cannot be used to overcome the prior art (e.g. see CCPA In re Lundenbera & Zuschlaa, 113, USPQ 530, 534 (1957)). For example, nowhere does applicant claim recite that the physical addresses must not subsequently determined when the logical addresses were mapped, must by directly used, or the like. The claim only recites :" ...registers that are relatively and absolutely addressable...". The fact that Panwar's relative addresses mapped to physical addresses does not means that the registers were not physically addressable. The registers in Panwar were physically addressable by the relative addresses. The claim does not recite whether the registers are directly addressable by physical addresses. Nevertheless, even if applicant has the dependent feature (claim 10) of an instruction field for directly specifying the absolute address register, Panwar taught that his register set was physically accessible by address register [304] and current pointer CWP [306] (see col.7, lines 48-50), and that the current window pointer [306] acted as an offset to address the registers in the register set (see col .7, lines 55-57). Panwar further taught a SPARC instruction [SAVE] allocated new register window and saved the prior register window by incrementing the current window pointer [CWP] (see col.7, lines 62-67, col.8, lines 1-5). Therefore, the CWP pointer (acted as offset address and physically accessing the registers) must be an operand or a field specified in the

> instruction format of the SAVE instruction so that the CWP pointer could be incremented by the instruction. Therefore, Pawar's SAVE instruction field [CWP], though not explicitly shown, was used for directly specifying the absolute address of a register. (Pages 4-5 of the Office Action).

Applicant disagrees.

Applicant's independent claim 1 recites "accessing, by a thread executing in the multithreaded processor, a register set organized into a plurality of windows of registers that are relatively and absolutely addressable per thread, wherein accessing absolutely any one of the relatively and absolutely addressable registers comprises providing an exact address of the register, the exact address specified in an instruction associated with the thread." Thus, when accessing registers using their absolute addresses, the exact address of the register is provided.

As explained in applicant's previous Amendment in Reply to Office Action, Panwar describes a processor that speculatively executes instructions that specify logical addresses, and a processor that converts the logical addresses to physical addresses (Abstract). Particularly, as shown in Panwar's FIG. 3, a register set is divided into windows having 32 registers. Different processes or programs executing on a processor 102 can allocate their own independent window (col. 7, lines 43-47). Panwar further describes that programs executing on a processor 102 access registers through a typical naming convention, such as r0, r1, r2, ..., r30 (col. 7, lines 51-54), and states:

> **The five-bit register addresses encoded in an instruction word specify the instruction's source registers and the destination register. These register specifiers are logical addresses that index registers within the current register window. (Col. 2, line 66 – col. 3, line 3)**

Programs executing on Panwar's apparatus therefore use relative (logical) addresses when specifying the desired registers that the programs are to access.

Panwar further explains:

> **An individual register within window 302 is physically accessible through register address 304 and current window pointer (CWP) 306. Because the window 302 has 32 registers, the register address 304 will be a 5-bit address. A program, however, would access the registers through a typical naming convention such as r0, r1, r2. . . r29, r30, and r31. In this sense, the current window pointer 306 acts as an offset to address the registers contained in the current window 302. While register file 300 has been shown having 128 registers, and window 302 has been shown as having 32**

> registers, it will be understood that the size of the register file and register windows is a matter of choice depending upon the needs of a particular application, and as such do not limit the present invention.
>
> In SPARC, certain instructions and architectural status registers relate to management of the register windows. As discussed above, a current window pointer (CWP) is maintained in a CWP register to track the current location of the window within the register file. A "SAVE" instruction allocates a new register window to the routine executing it, and saves the prior register window by incrementing the CWP register. A "RESTORE" instruction restores the previous register window (i.e., the register window saved by the last SAVE instruction executed by the current process) by decrementing the CWP register. (Panwar's col. 7, line 48, to col. 8, line 5)

Thus, the CWP holds the current location of the register window used by the currently executing routine or program, and that location is used as an offset value for computing the physical addresses of the registers being accessed. A "SAVE" instruction causes the offset value in the CWP to change by incrementing the current value held in the CWP. A RESTORE instruction causes the offset value in the CWP register to be decremented. At no point does Panwar describe or suggest that either the SAVE instruction or the RESTORE instruction specify an actual physical address of any register.

Therefore, since Panwar describes that register-access instructions use relative addressing, and since there is no suggestion in Panwar that the SAVE and/or RESTORE instructions specify physical offset addresses, it follows that Panwar does not describe or suggest at least "wherein accessing absolutely any one of the relatively and absolutely addressable registers comprises providing an exact address of the register, the exact address specified in an instruction associated with the thread," as required by applicant's independent claim 1.

As noted above, the examiner also used the Cheng reference to maintain his rejections of old claim 1. Applicant first notes that the examiner did not use the Cheng reference to reject any of the other claims. Thus, applicant contends that this constitutes an implicit admission by the examiner that the other pending claims of the above-identified application are not anticipated by Cheng. Since amended independent claim 1 includes features similar to those recited in old claims 9 and 10, which the examiner did not reject on the basis of the Cheng reference, for this reason alone applicant considers amended claim 1 to be patentable over Cheng.

Moreover, with respect to the examiner argument vis-à-vis Cheng, the examiner stated:

13. As to c), Chan taught registers parts may be addressed by a calling procedure (see Section 3 Page 58). The call statement itself is a relative addressing because it is the programming statement by programmer. The call statement was also directed to a physical address because it needs to know the target location of registers; (Pages 5-6 of the Office Action)

Applicant disagrees.

Cheng describes a compilation technique that supports pipelining and multithreading. While Cheng briefly mentions using register windows and organizing registers into different parts that may be accessed by either a calling procedure, a child procedure (i.e., callee), or other windows (see section 3, page 58), Cheng does not disclose how such registers are accessed.

Moreover, like Panwar, Cheng also uses a current window pointer (CWP) in its implementation of the register windows. The use of a CWP thus suggests the use of relative addressing to access Cheng's register windows. Indeed, the examiner admitted that "[t]he call statement itself is a relative addressing because it is the programming statement by programmer." Applicant, however, fails to find any statement or suggestion in Cheng that supports the examiner's statements that "[t]he call statement was also directed to a physical address because it needs to know the target location of registers." Also, Cheng describes that the SAVE and RESTORE instruction used by SPARC processor cause the value CWP to respectively decrease and increase, but there is no indication that those instructions, or any other register-access instruction, specify a register's physical address.

Accordingly, applicant contends that Cheng neither describes nor suggests at least "wherein accessing absolutely any one of the relatively and absolutely addressable registers comprises providing an exact address of the register, the exact address specified in an instruction associated with the thread," as required by applicant's independent claim 1.

Thus, since neither Panwar nor Cheng disclose or suggest, alone or in combination, at least the feature of "wherein accessing absolutely any one of the relatively and absolutely addressable registers comprises providing an exact address of the register, the exact address specified in an instruction associated with the thread," applicant's independent claim 1 is therefore patentable over the cited art.

Claims 2-8 and 10 depend from independent claim 1 and are therefore patentable for at least the same reasons that independent claim 1 is patentable.

Independent claim 12 recites "a register set that is organized into a plurality of windows of registers that are relatively and absolutely addressable by an executable thread, wherein any one of the registers of the register set is configured to be accessed absolutely by providing an exact address of the register, the exact address specified in an instruction associated with the executable thread." At least this feature is not disclosed by the cited art for reasons similar to those provided with respect to applicant's independent claim 1. Accordingly, independent claim 12 is patentable over the cited art.

Claims 13-20 depend from independent claim 12 and are therefore patentable for at least the same reasons as independent claim 12.

Independent claim 21 recites "wherein the instructions that cause the processor to access absolutely any one of the relatively and absolutely addressable registers comprise instructions that, when executed, cause the processor to provide an exact address of the register, the exact address specified in an instruction associated with the thread." At least this feature is not disclosed by the cited art for reasons similar to those provided with respect to applicant's independent claim 1. Accordingly, independent claim 21 is patentable over the cited art.

It is believed that all the rejections and/or objections raised by the examiner have been addressed.

Canceled claims, if any, have been canceled without prejudice or disclaimer.

Any circumstance in which the applicant has (a) addressed certain comments of the examiner does not mean that the applicant concedes other comments of the examiner, (b) made arguments for the patentability of some claims does not mean that there are not other good reasons for patentability of those claims and other claims, or (c) amended or canceled a claim does not mean that the applicant concedes any of the examiner's positions with respect to that claim or other claims.

In view of the foregoing remarks, applicant respectfully submits that the application is in condition for allowance and such action is respectfully requested at the examiner's earliest convenience.

Attorney's Docket No.: 10559-310US1 / Intel Corporation P9631

Applicant : Gilbert Wolrich et al.
Serial No. : 10/070,091
Filed : February 27, 2002
Page : 12 of 12

No fee is believed due. Please apply any charges to deposit account 06-1050, referencing attorney docket 10559-310US1.

Respectfully submitted,

Date: April 6, 2006

Ido Rabinovitch
Attorney for Intel Corporation
Reg. No. L0080

Fish & Richardson P.C.
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21304093.doc